PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/00, H04K 1/00

A1

(11) International Publication Number: **WO 98/31124**

(43) International Publication Date: 16 July 1998 (16.07.98)

(21) International Application Number: PCT/US98/01117

(22) International Filing Date: 6 January 1998 (06.01.98)

(30) Priority Data:
08/782,479 10 January 1997 (10.01.97) US

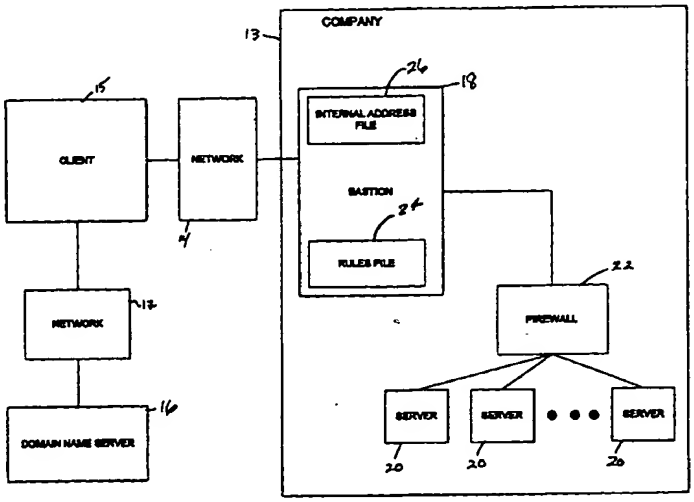
(71)(72) Applicants and Inventors: HANSON, Gordon, L. [US/US]; 16613 S.E. 254th Place, Kent, WA 98042 (US). HANSON, Kevin, L. [US/US]; 16613 S.E. 254th Place, Kent, WA 98042 (US).

(74) Agent: SMITH, Michael, S.; Christensen O'Connor Johnson & Kindness, Suite 2800, 1420 Fifth Avenue, Seattle, WA 98101 (US).

(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: REVERSE PROXY SERVER



(57) Abstract

A method and system for securely accessing servers over an internetwork (14). Each server includes a processor and a memory. A first server (18) outside a company's firewall (22) connects the company to the internetwork. A user or client (15) sends a data packet with a server name to a second server (20) identified by the server name and located within the company's firewall. The location address of the first server is retrieved according to the domain/server name, a connection is made with the first server according to the retrieved location address of the first server and the data packet with domain/server name in the sent data packet to a list of at least one internal address, wherein the at least one internal address of the list identifies the location of the second server. If an internal address is found to match the server name, the first server sends the packet to the internal address.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

REVERSE PROXY SERVER

Field of the Invention

This invention relates to client/server computer communication over an internetwork system and, more particularly, to improved access of firewall protected servers.

Background of the Invention

Communication networks are well-known in the computer communications field. By definition, a network is a group of computers and associated devices that are connected by communications facilities or links. Network connections can be of a permanent nature, such as via cables, or can be of a temporary nature, such as connections made through telephone or other communication links. Networks vary in size, from local area network (LAN) consisting of a few computers and related devices, to a wide area network (WAN) which interconnects computers and LANs that are geographically dispersed. An internetwork, in turn, is the joining of multiple computer networks, both similar and dissimilar, by means of gateways or routers that facilitate data transfer and conversion from various networks. A well-known abbreviation for internetwork is "Internet." As currently understood, the capitalized term "Internet" refers to the collection of networks and routers that use a Transmission Control Protocol/Internet Protocol (TCP/IP) to communicate with one another.

A representative section 10 of the Internet is shown in FIGURE 1 (Prior Art) in which a plurality of LANs 12 are interconnected by routers 11. The routers 11 are generally special purpose computers used to interface one LAN to another. Communication links within the LANs may be twisted wire pair, or coaxial cable,

while communication links between networks may utilize 56Kbps analog telephone lines, 1Mbps digital T-1 lines and/or 45Mbps T-3 lines. It will be appreciated that the Internet comprises a vast number of such interconnected networks and routers and that only a small representative section of the Internet is shown in FIGURE 1.

The rapid growth and development of the Internet has made the Internet an important business tool. A company's primary concern when connected to the Internet is security. The advantages of using the Internet can immediately be nullified by the possibility of internal company computers being compromised by an external entity. Ruinous results occur if data is stolen or computers are infected by viruses. The immediate solution created is the firewall, a filter which can work either at the circuit or software level. This solution has been widely accepted by corporations and, with proper care and administration, effectively allows a company to securely utilize the Internet. The firewall's primary fault is that it blocks communication in both directions, input to the internal network of the company and output to the Internet.

Proxy servers act as relay stations between an internal network and the Internet for communication requests initiated inside the company's network, thus helping solve the known problem. The fundamental security basis to a proxy server is the trust of all internal computers. It is assumed that any communication requested by internal computers is not going to compromise the security of internal computers. This is a valid assumption in most cases. Unfortunately, the opposite of this assumption is not valid. Computers on the Internet at large cannot be trusted without elaborate authentication and encryption, see U.S. Patent No. 5,550,984. This poses a major problem to companies on the Internet which, selectively, want to share internal company resources on the Internet.

For example, company A has a printer attached to their own network. Whenever someone wants to print a job to this printer, they just print over the network to the printer's server. Company A is also attached to the Internet with a firewall protecting their valuable computer secrets. Now, company A decides that it would like to allow a few allies, company B, C, and D, to be able to print confidential documents directly to company A's printer. Instead of having to ship printed contracts via courier or U.S. Mail, the allies would securely send the document directly to the printer inside company A's network. There are a few immediate solutions to accomplish what company A wants. A first solution would require the physical movement of the printer to a new network which is open to the Internet. Another solution would require that company A's firewall be opened to allow

connection from the Internet to the printer. Both solutions would require a new InterNIC valid IP address, something which recently is increasingly in short supply, and each computer inside company A would have to be reconfigured to use the new IP address for the printer.

Accordingly, there is a need for secure two-way data communication over the Internet. The present invention is directed to providing such communication security.

Summary of the Invention

In accordance with this invention, a method and system for securely accessing servers over an internetwork are provided. Each server includes a processor and a memory. The location address of a first server is determined. Then, the user sends a data packet with a server name to the first server according to the determined location address, wherein the server name is associated with a second server connected to the internetwork through the first server. The first server compares the server name in the sent data packet to at least one internal address, wherein the at least one internal address identifies the location of said second server. If an internal address is found to match the server name, the packet is sent to the found internal address.

In accordance with other aspects of the present invention, a client or user transmits a server name to a third server prior to sending said data packet. The user receives a location address from the third server in response to the transmitted server name, wherein the location address is the address identifying a first server associated with the server name and connected to the internetwork.

In accordance with still other aspects of the present invention, the third server is a server name server and the first server is a web server.

In accordance with further aspects of the present invention, communication between the first server and the second server is performed through a firewall.

In accordance with yet other aspects of the present invention, the list of at least one internal address is encrypted. The encrypted list is decrypted then compared to the server name of the received data packet. The decrypted list is deleted at the completion of the comparison.

In accordance with still further aspects of the present invention, the user receives an error reply if no match was found in the compare of the server name with the list of internal addresses or if a reply data packet from the second server fails to be sent from the first server.

Brief Description of the Drawings

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a schematic diagram illustrating a network, such as the Internet;

FIGURE 2 is a block diagram of a preferred embodiment of a system configuration of the present invention;

FIGURE 3 is an example system functioning according to the invention of FIGURE 2;

FIGURES 4 and 5 are flow diagrams illustrating the steps for executing the invention shown in FIGURE 2; and

FIGURES 6 and 7 are flow diagrams illustrating an example packet request and reply according to steps shown in FIGURES 4 and 5.

Detailed Description of the Preferred Embodiment

As shown in FIGURE 2, a network system provides secure bi-directional data packet communication between a requester or client 15 and servers 20 protected by a firewall according to the present invention. A company 13 includes a number of internal servers 20 protected by a firewall 22 and a bastion server 18 connected to a network 14. Client 15 also connects to the network 14, and includes a domain name server 16 or connects to a domain name server 16 over a network 17. Networks 14 and 17 may be the same network. Client 15 can be a separate entity from company 13 or a separate entity in a different department of company 13. Examples of networks 14 or 17 are the Internet and an intranet.

FIGURE 3 illustrates an example of computer components and peripheral devices that may be used in place of like numbered block diagram components shown in FIGURE 2. As shown in FIGURE 3, a computer 20a, file server 20b, printer 20c and facsimile device 20d are examples of computers or peripheral devices that can act as or include a server 20 connected within Company A's firewall 22 (FIGURE 2). Company A's system 13a includes a bastion server 18a coupled between the network 14a and the Company A's firewall 22a. The firewall 22a provides discrete access to the servers 20a-d located inside Company A.

As shown in FIGURE 2, the bastion 18 is a server that includes a processor and memory like typical servers, but also includes an internal address file 26 and a rules file 24 stored in memory. The internal address file 26 includes the internal

addresses for each of the servers 20 the company wishes to allow external access to through the firewall 22. Any data packet sent by a client specifically identifying a server 20 located behind another entity's (company 13) firewall 22, first goes to the client's domain name server 16 for retrieval of the IP address assigned to the bastion 18 that protects the destination server 20. The IP address is the bastion 18's location on the network 14. The client may include a domain name server, thereby reducing the step of having to retrieve from an external domain name server. The data packet is then sent to the retrieved destination IP address. The rules file 24 provides a predefined set of rules for maintaining secure communication of data packets passing in both directions through bastion server 18. Bastion 18 limits communication with a single server 20 within firewall 22 without putting the company's system 13 at risk to rogue clients or unauthorized requests. The bastion 18 can perform this service because the internal address file 26 is the only location where internal IP addresses are stored. The limited secure communication performed by the bastion 18 is described in more detail below with reference to the example illustrated in the Tables.

FIGURE 4 is an information flow diagram according to the invention shown in FIGURE 2. At block 40, the client uses a domain (server) name to request the IP address of a bastion coupled to the server identified by the server name through a firewall. This request can be accomplished by various methods known in the art. Supplying the server name directly to a domain name server is one method and performing an automatic request from a domain name server at the time the client sends a data packet is another. In order to maintain a certain level of security, the client receives the server name from the company under confidence prior to execution of any transactions. At block 41, if the client does not know the IP address assigned to the bastion associated with the server name or have the ability to force a packet with the server name to the IP address assigned the bastion, the domain name server (DNS) retrieves the IP address associated with the requested server name, see block 42. Within the DNS, all names of accessible servers within a company's firewall are stored with direct reference to the company's IP address (IP address of the company's bastion). Upon finding the requested server name, the DNS returns the directly referenced IP address, see block 44. Thus, a company only requires only a single IP address for identifying all the servers within its system. This feature of needing a limited number of IP addresses is important because InterNIC, the Internet IP address controlling entity, is limiting the number of IP addresses because available

addresses are running out. If a first DNS fails to include the requested information, other DNSs are accessed until the information is found or its is determined the information does not exist. DNSs are tied together in a distributed database system which provides a system of interconnected DNSs with databases that can automatically update there internal files according to the frequency of particular requests.

Now that the client has the company's IP address, the client establishes a connection with the bastion, see block 46. After a connection is made, the data packet, which includes the server name, is sent to the bastion, see block 48. The bastion then compares the server name to the internal address file, see block 52. At decision block 54, the bastion determines whether a match exists between the server name and an internal address located in the internal address file. If no match is found, the bastion sends a reply to the client indicating that the request packet cannot be delivered because the requested server is not listed or no longer listed in the internal address file within the company or the request is incorrect for some other predefined reason, see block 56. If a match was found, the received packet is checked against rules contained within the rules file, see decision block 58. If the received data packet fails to pass any of the predefined rules in the rules file, a reply is sent to the client indicating so, see block 56. However, if the received data packet passes all the rules contained within the rules file, a connection is made between the client and the server associated with the matched internal IP address and the data packet is delivered to the server. The rule checks include certain security- programs that operate upon received data packets and, particularly, data packets that are or include programs. Some unique rules and rules in the form of programs are described in more detail below with respect to Table 3.

FIGURES 6 and 7 illustrate an example of two-way packet communication between a client and a firewall protected server over the Internet. The client's and bastion's IP addresses are 245.23.12.3 and 124.12.32.1, respectively. The client's goal is to establish two way data packet communication with a server protected by a firewall and connected to the Internet through a bastion. As shown in FIGURE 6, at block 80, the client sends a request packet out on the Internet to the destination server, fuji1.hde.com.

@	IN	SOA	bastion.hde.com. root.bastion.hde.com (
		8	;Serial
		21600	;Refresh 6 hours
		900	Retry 15 minutes
		172800	;Expire 48 hours
		3600)	;Minimum 1 hour
	IN	NS	bastion.hde.com
	IN	MX	1 bastion.hde.com
ns	IN	CNAME	bastion.hde.com
localhost		IN	A 127.0.0.1
bastion		IN	A 124.12.32.1
laser_printerA		IN	A 124.12.32.1
file_serverA		IN	A 124.12.32.1
jp_www		IN	A 124.12.32.1
fuji1		IN	A 124.12.32.1

Table 1 - DNS zone file.

At block 82, a domain name server (DNS) searches for the domain name zone file associated with the server name that ends with hde.com. The DNS runs the DNS server program called "named" that reads the zone file, shown in Table 1. The zone file provides the translation of the requested server name, fuji1, into the IP address, 124.12.32.1, for the bastion connecting the server identified by the server name to the Internet, see block 84. If no IP address matches the server name, fuji1, an alternate DNS is accessed for locating an IP address associated with the server name. The alternate DNS is identified in the DNS zone file by the code line: "IN NS bastion.hde.com." NS is name server. At block 86, the client establishes a connection to the bastion with IP address 124.12.32.1. Port 80 of the bastion is the receiving port and the client's connecting port depends upon other client port connections. In this example the client connecting port is port 80.

After communication is established, the bastion receives the data packet(s) from the client at block 88. At block 90, the server name, fuji1.hde.com, and the data packet are transmitted to the bastion. A program operating within the bastion

provides the necessary coordination for the received data packet. The server name, fuji1.hde.com, is first compared to a decrypted internal address file, internal.conf, see Table 2. The internal IP address file is decrypted into active memory upon data packet reception and the decrypted internal IP address file is deleted from memory upon completion of the comparison. Since the decrypted internal IP address file appears in active memory, i.e., RAM, for a very short period of time, it is virtually impossible for one to access this information in an attempt to discover the internal addresses. The internal address file presents a list of domain names and corresponding internal IP addresses. These internal addresses are known only to the bastion.

jp_www.hde.com = 204.96.95.251
fuji1.hde.com = 204.96.95.95
ftp.hde.com = 204.96.95.251

Table 2 - internal IP address file (internal.conf) after decryption.

At block 92, the server name, fuji1.hde.com, is found to match an internal server with address 204.96.95.95. The data packet then passes the check against the set of rules within the rules file at block 94, see Table 3. Then, a connection from the client through the bastion to the internal server is established through port 80 of the internal server and a predefined output port on the bastion, at block 96. After connection is established, the internal server receives and processes the client's request packet at block 98.

FIGURE 7 illustrates an example of a reply packet sent from the internal server through the bastion back to the client. At block 100, the internal server first sends the reply packet to the bastion. In this example, the reply data packet is a reply to the data packet received from the client. The reply packet may also be an unsolicited data packet. At block 102, the reply data packet passes a check against a set of rules for outgoing data packets. Similar to checks of received data packets, the outgoing data packet check is performed within the bastion. And finally at block 104, the outgoing data packet is sent to the client with IP address 245.23.12.3.

```
Destination IP: 204.95.95.0-94
FILE: viruses.dat
FILE: ps_error.dat
Flags: 0110001101
URL: fuji1.hde.com/target.html

JAVA Checks:
 signature.class
 security.class
 test.class

ActiveX Checks:
 security.ocx
```

Table 3 - rules file.

Table 3 illustrates an example set of rules for checking data packets passing through the bastion. As shown in Table 3, the code line "Destination IP: 204.95.95.0-94" is an IP address limiter rule. If an internal address discovered in the internal address file check of a received data packet is within the range of 204.95.95.0 through 204.95.95.94, the received data packet is denied access. The limiter rule may also be used to present a range of allowable IP addresses. By using a IP address limiter, a system operator can limit access to specific internal servers. Another check performed is a file compare of the received data packet against prespecified files, as indicated by the code lines beginning with "FILE:". As shown in Table 3, the received data packet is checked against the indicated files "viruses.dat" and "ps_error.dat." These files compare the data packet to known viruses and data errors. If the data packet fails to pass the check with either one of these files, the data packet is refused and destroyed, because it most likely contains a virus. Similar file compare programs may also be used in the line "FILE:". The code line "Flags:" prevents so called "Christmas tree" data packets from being relayed to the internal server by the bastion and causing problems with the internal server. As shown in Table 3, a received packet that exhibits the flags 0110001101 in its header will not be processed by the bastion.

The "URL:" code line identifies all URL addresses the internal server is denied access to. If an internal server attempts to address an address contained within the

"URL:" code line, the access is denied and an error message is returned to the client explaining the failed transaction. In the case that any one of the rules are broken and a specific service is denied, a log file is maintained within the bastion to record information regarding the denied transaction. The log file may include such things as why the denial was made, who it was from, and what was the destination of the packet. It can be appreciated to those of ordinary skill in the art that the rules file can include very specific rules relating to the type of system in which it is being used.

The "JAVA Checks:" code lines indicate JAVA class files that execute if the bastion receives a JAVA applet as or in a data packet. The JAVA class files listed are "signature.class", "security.class" and "test.class". Each perform specific checks of received JAVA applets. The "security.class" program ensures that the data within the data packet going to the client or internal server is not destructive for the intended recipient. The "security.class" program performs security operations similar to that performed by a complete, secure JAVA virtual machine. "Security.class" performs these protective illegal operation overrides by attaching itself to the applet being sent in the data packet. When the applet intended for the recipient is run at the destination client or server, "security.class" is run simultaneously. Since every system operable with JAVA applets includes a JAVA virtual machine, "security.class" sometimes performs redundant security checks to those performed by a complete JAVA virtual machine, thus protecting against a bad or incomplete JAVA virtual machine. The "signature.class" program performs a certification operation similar to those provided by VeriSign Corporation's programs. The "signature.class" program authenticates received JAVA applets by adding a signature program and/or time stamp to the received applet. When the applet intended for the destination client or server is run, the signature and/or time stamp program is also run to insure that in between transmission from the client server and the destination, the applet was not tampered with or altered in any manner. If the applet will not run and the destination client or server will be informed that the applet received was invalid.

The "test.class" program performs two checks of the applet being transmitted through the bastion server. First, it compares the applet against known virus code, similar to a virus checking program. Next, it communicates with a secure Java virtual machine located on the bastion. The Java virtual machine on the bastion executes the applet intended for the destination client or server. If any illegal operations are attempted by the applet, the Java virtual machine informs the "test.class" program and

the "test.class" program relays to the a program on the bastion that the Java applet is not secure and should not be transmitted to the destination client or server.

The "ActiveX Checks:" section includes the program "security.ocx". The "ActiveX Checks:" section is similar to the "Java Checks:" section except it applies to data packets which are ActiveX programs which are intended to be run on the destination client or server. The "security.ocx" program attaches to an ActiveX program destined for a client or server. This program is run at the destination client or server and behaves similar to a common virus checking program. It monitors the execution of the ActiveX program that is running on the destination client or server. If the ActiveX program attempts an operation which is destructive to the host client or server, the "security.ocx" stops execution and warns the client or server user that the ActiveX program attempted an illegal operation.

It can be appreciated to those of ordinary skill in the art that the attachment of security programs and the examination of the data packets at the bastion or the destination client or server can be extended to provide various levels of security and protection against various types of data being transmitted. It also can be appreciated that resources are reserved at the bastion, if the execution of a security program is performed at the destination client or server.

While the preferred embodiment of the invention has been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the invention.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method for securely sending data to servers over an internetwork, wherein each server comprises a processor and a memory, said method comprising:

determining the location address of a first server;

sending a data packet with a server name to said first server according to said determined location address, wherein said server name being associated with a second server connected to the internetwork through said first server;

comparing within said first server said server name in said sent data packet to a list of at least one internal address, wherein said at least one internal address of said list identifies the location of said second server; and

if an internal address is found to match the server name, sending the packet to said found internal address.

2. The method of Claim 1, further comprising:

transmitting a domain name to a third server prior to sending said data packet; and

receiving a location address from said third server in response to said transmitted domain name, wherein said location address being the address identifying said first server associated with said domain name and connected to the internetwork.

3. The method of Claim 2, wherein said third server being a domain name server.

4. The method of Claim 1, wherein said first server being a web server.

5. The method of Claim 1, wherein communication between said first server and said second server connected to the internetwork through said first server is performed through a firewall.

6. The method of Claim 1, wherein said list of at least one internal address is encrypted.

7. The method of Claim 1, wherein comparing further comprises:

decrypting said encrypted list;

comparing said decrypted list to said server name; and

deleting said decrypted list at completion of said comparison.

8. The method of Claim 1, further comprising generating an error message, if no match was found in said compare of said server name with the list of at least one internal address.

9. The method of Claim 1, further comprising:

performing at least one of a signature, a security and a test check of the received data packet within said first server; and

sending said data packet to said found internal address, if all performed checks are successful.

10. The method of Claim 1, further comprising:

tagging a security check program onto the received data packet, if a specific program exists within the received data packet; and

performing the security check program simultaneously with the specific program of the received data packet at said second server.

11. A system for securely accessing servers over an internetwork, wherein each server comprises a processor and a memory, said system comprising:

a means for determining the location address of a first server;

a means for sending a data packet with a server name to said first server according to said determined location address, wherein said server name being associated with a second server connected to the internetwork through said first server; and

a means for comparing within said first server said server name in said sent data packet to a list of at least one internal address, wherein said at least one internal address of said list identifies the location of said second server, and if an internal address is found to match the server name, sending the packet to the internal address.

12. The system of Claim 11, further comprising:

a means for transmitting a domain name to a third server prior to sending said data packet; and

a means for receiving a location address from said third server in response to said transmitted domain name, wherein said location address being the address identifying said first server associated with said domain name and connected to the internetwork.

13. The system of Claim 12, wherein said third server is a domain name server.

14. The system of Claim 11, wherein said first server is a web server.

15. The system of Claim 11, wherein communication between said first and second server is performed through a firewall.

16. The system of Claim 11, wherein said list of at least one internal address is encrypted.

17. The system of Claim 11, wherein said first server further comprises:

a means for decrypting said encrypted list;

a means for determining from said decrypted list the location of the server not directly connected to the internetwork with said received server name; and

a means for deleting said decrypted list at completion of said determination.

18. The system of Claim 11, wherein said first server further comprises transmitting an error message to said third server if no internal address was determined to match said server name of the received data packet.

19. The system of Claim 11, further comprising:

a means for performing at least one of a signature, a security and a test check of the received data packet within said first server; and

a means for sending said data packet to said found internal address, if said all performed checks are successful.

20. The system of Claim 11, further comprising:

a means for tagging a security check program onto the received data packet, if a specific program exists within the received data packet; and

a means for performing the security check program simultaneously with the specific program of the received data packet at said second server.

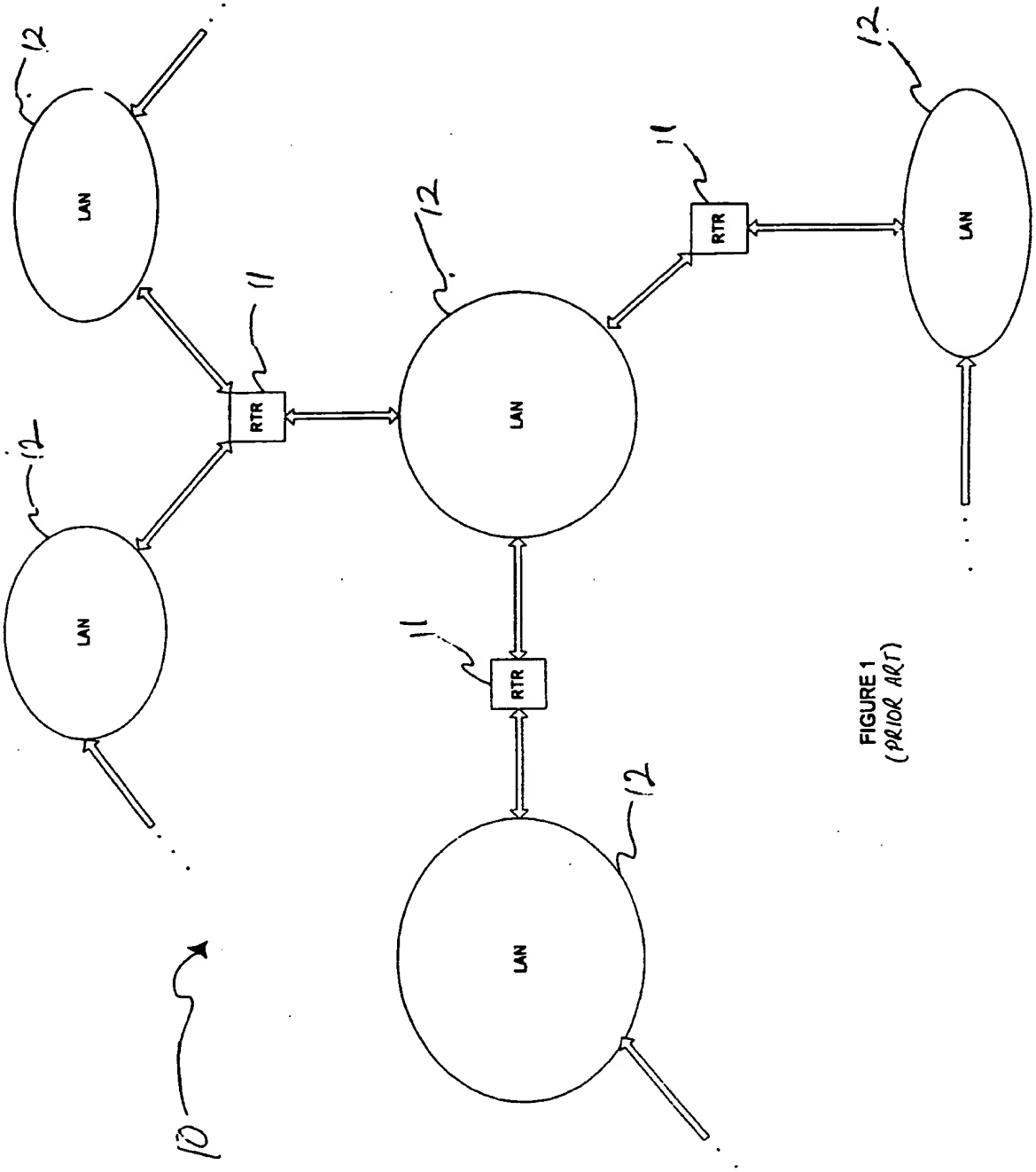


FIGURE 1
(PRIOR ART)

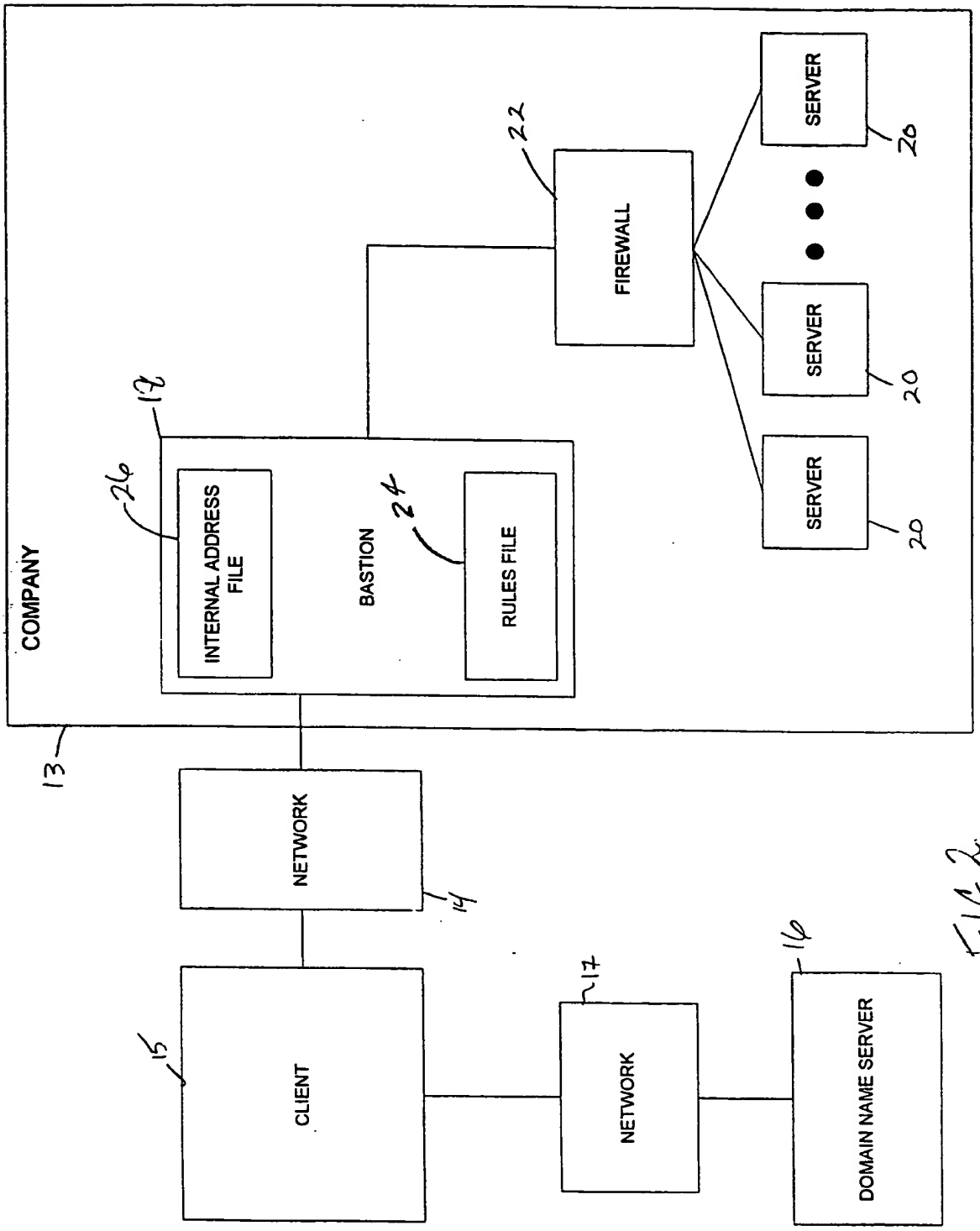

COMPANY
13
INTERNAL ADDRESS FILE
26
BASTION
17
RULES FILE
24
FIREWALL
22
SERVER
20
SERVER
20
SERVER
20
NETWORK
14
CLIENT
15
NETWORK
17
DOMAIN NAME SERVER
16

FIG 2

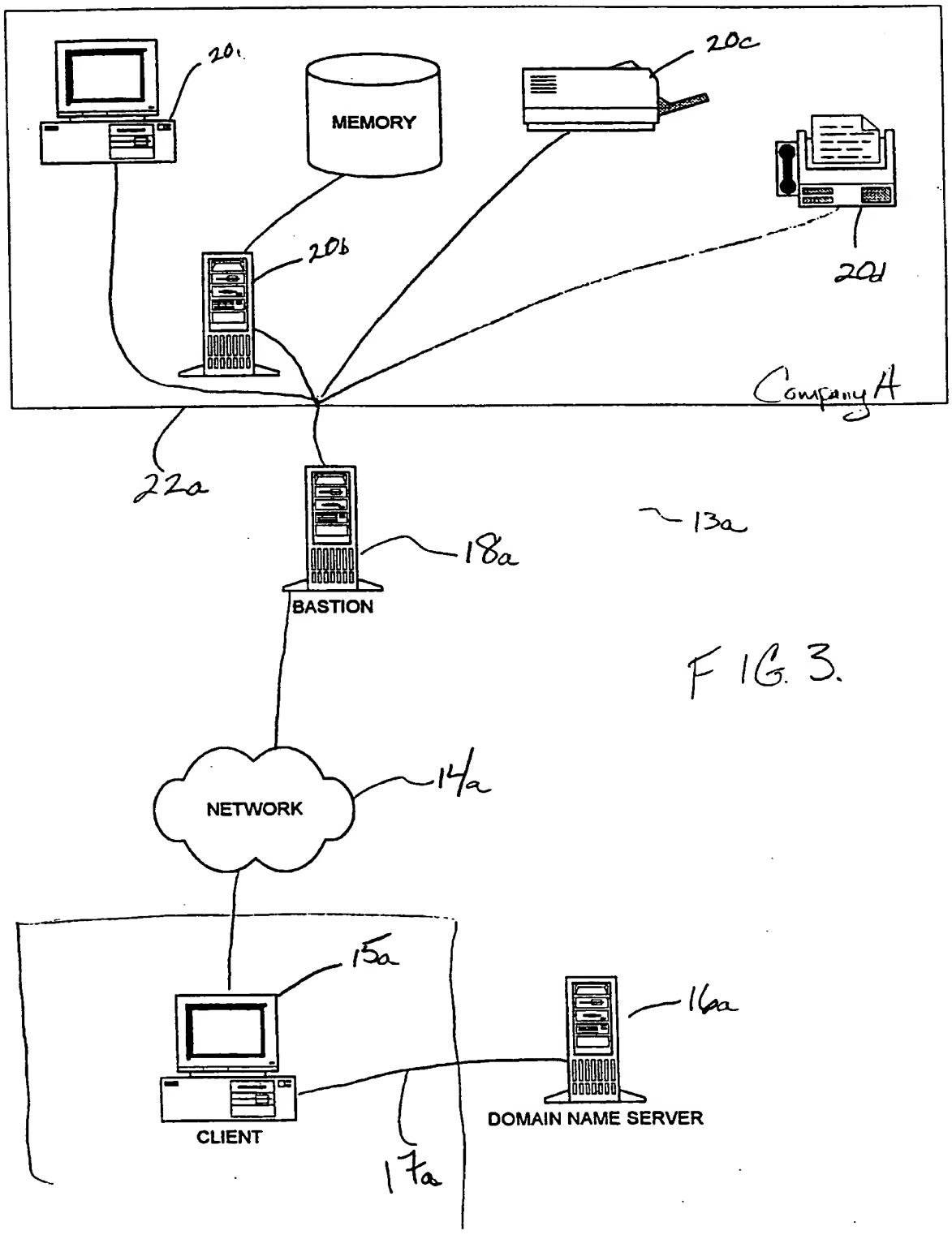

20a
MEMORY
20c
20b
20d
Company A
22a
13a
18a
BASTION
NETWORK
14a
15a
16a
CLIENT
DOMAIN NAME SERVER
17a

FIG. 3.

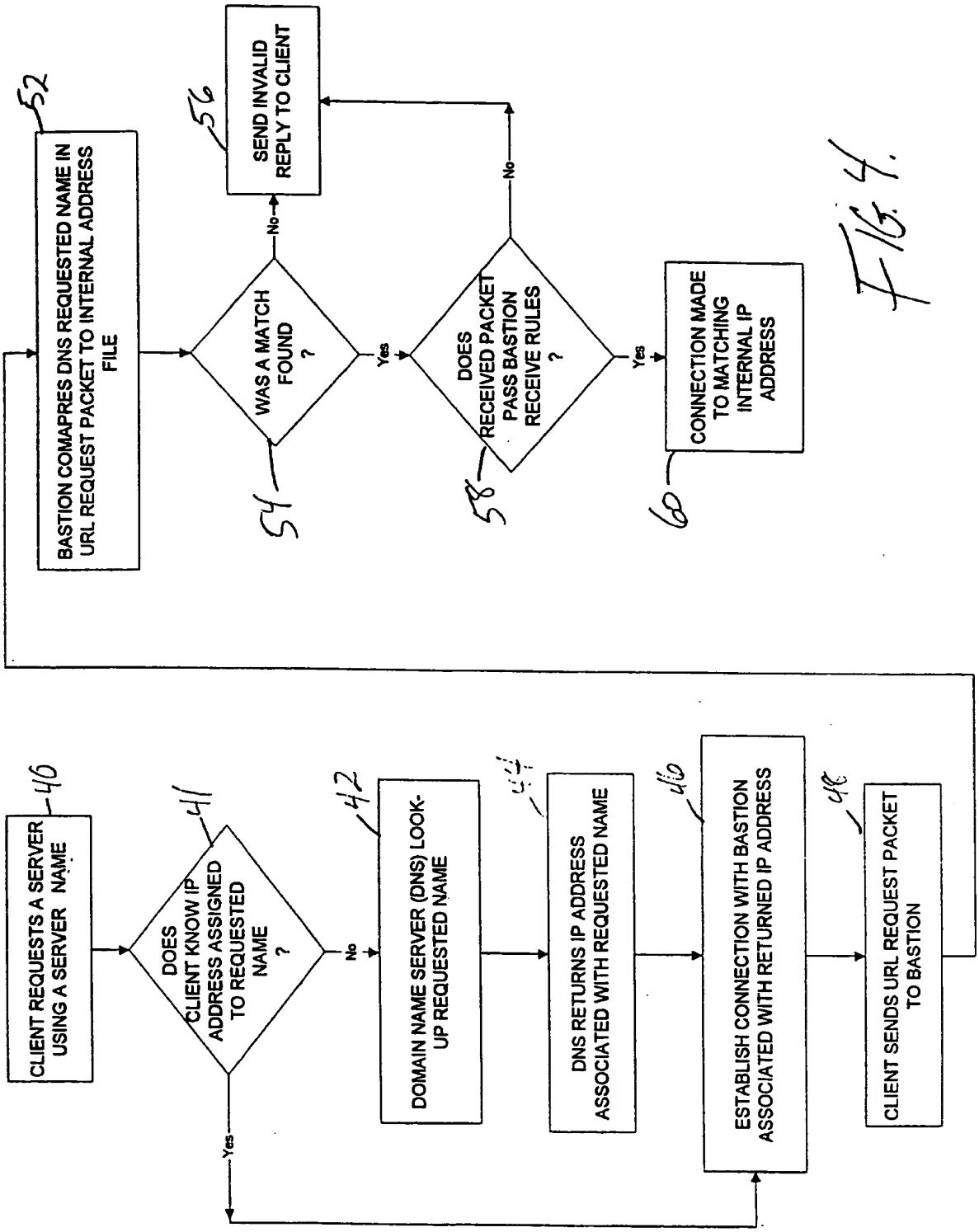

CLIENT REQUESTS A SERVER USING A SERVER NAME
46
DOES CLIENT KNOW IP ADDRESS ASSIGNED TO REQUESTED NAME ?
41
No
Yes
DOMAIN NAME SERVER (DNS) LOOK-UP REQUESTED NAME
42
DNS RETURNS IP ADDRESS ASSOCIATED WITH REQUESTED NAME
44
ESTABLISH CONNECTION WITH BASTION ASSOCIATED WITH RETURNED IP ADDRESS
46
CLIENT SENDS URL REQUEST PACKET TO BASTION
48
BASTION COMAPRES DNS REQUESTED NAME IN URL REQUEST PACKET TO INTERNAL ADDRESS FILE
52
WAS A MATCH FOUND ?
54
No
SEND INVALID REPLY TO CLIENT
56
Yes
DOES RECEIVED PACKET PASS BASTION RECEIVE RULES ?
58
No
Yes
CONNECTION MADE TO MATCHING INTERNAL IP ADDRESS
60

FIG. 4.

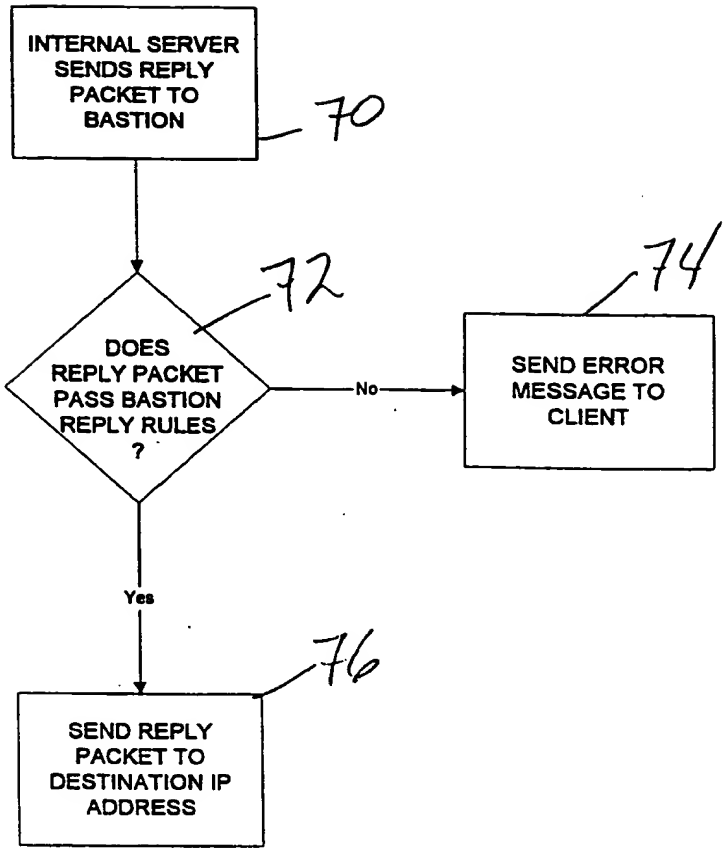

INTERNAL SERVER SENDS REPLY PACKET TO BASTION
70
DOES REPLY PACKET PASS BASTION REPLY RULES ?
72
No
SEND ERROR MESSAGE TO CLIENT
74
Yes
SEND REPLY PACKET TO DESTINATION IP ADDRESS
76

FIG. 5

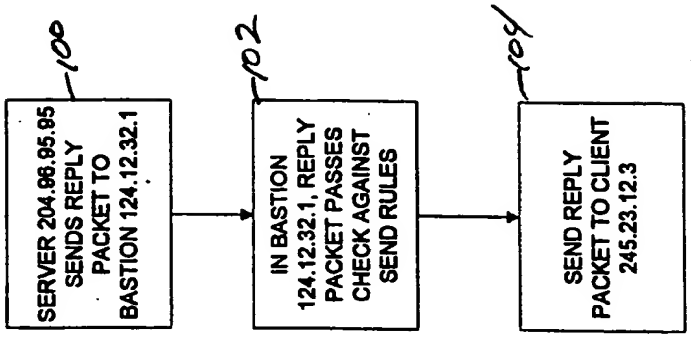

100
SERVER 204.96.95.95 SENDS REPLY PACKET TO BASTION 124.12.32.1
102
IN BASTION 124.12.32.1, REPLY PACKET PASSES CHECK AGAINST SEND RULES
104
SEND REPLY PACKET TO CLIENT 245.23.12.3

FIG. 7.

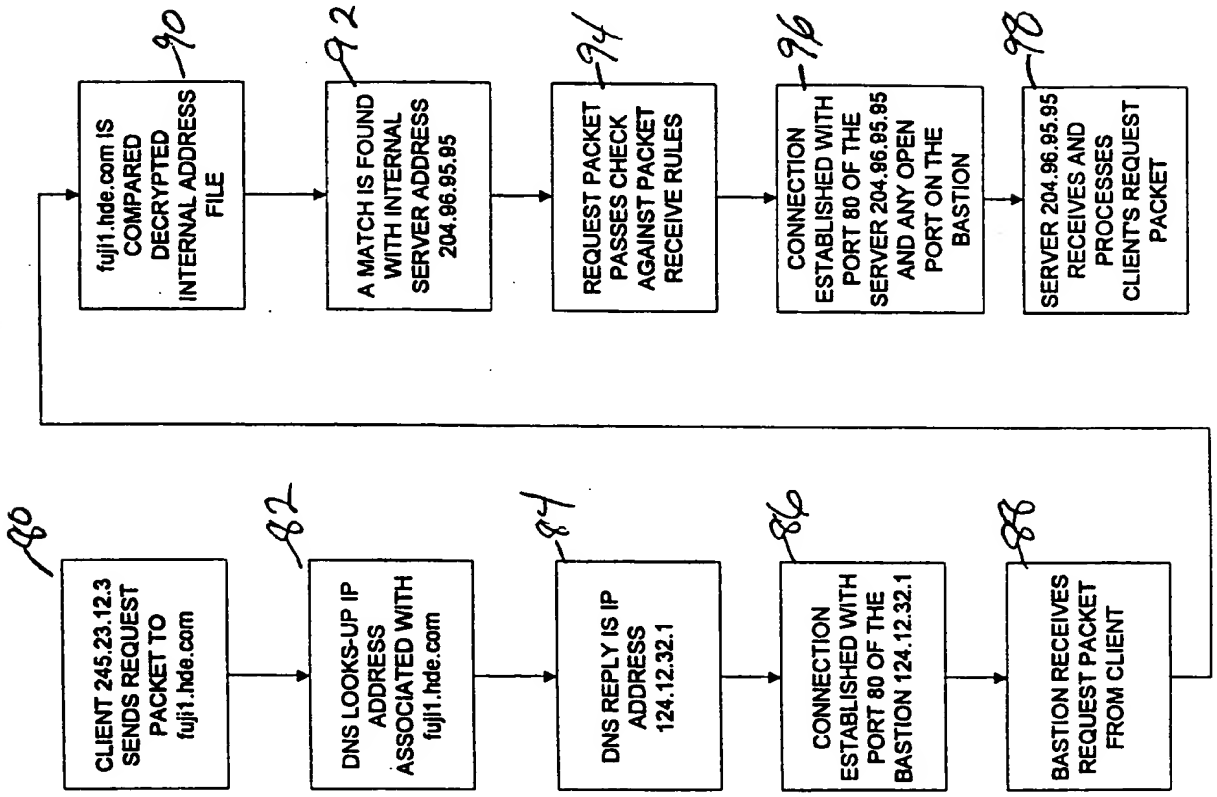

80
CLIENT 245.23.12.3 SENDS REQUEST PACKET TO fuji1.hde.com
82
DNS LOOKS-UP IP ADDRESS ASSOCIATED WITH fuji1.hde.com
84
DNS REPLY IS IP ADDRESS 124.12.32.1
86
CONNECTION ESTABLISHED WITH PORT 80 OF THE BASTION 124.12.32.1
88
BASTION RECEIVES REQUEST PACKET FROM CLIENT
90
fuji1.hde.com IS COMPARED DECRYPTED INTERNAL ADDRESS FILE
92
A MATCH IS FOUND WITH INTERNAL SERVER ADDRESS 204.96.95.95
94
REQUEST PACKET PASSES CHECK AGAINST PACKET RECEIVE RULES
96
CONNECTION ESTABLISHED WITH PORT 80 OF THE SERVER 204.96.95.95 AND ANY OPEN PORT ON THE BASTION
98
SERVER 204.96.95.95 RECEIVES AND PROCESSES CLIENT'S REQUEST PACKET

FIG. 6.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/01117

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) H 04 L 9/00; H 04 K 1/00
US Cl 380/49, 23, 25
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/49, 23, 25

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,583,940 A (VIDRASCU et al.) 10 December 1996, the whole document.	1-20
X	US 5,550,984 A (GELB) 27 August 1996, the whole document.	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 26 MAY 1998	Date of mailing of the international search report 25 JUN 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer HRAYR A. SAYADIAN Telephone No. (703) 306-4169

Form PCT/ISA/210 (second sheet)(July 1992)*